Language Acquisition with Echo State Networks: Towards Unsupervised Learning

Thanh Trung Dinh, Xavier Hinaut

Language Acquisition with Echo State Networks: Towards Unsupervised Learning

Thanh Trung Dinh
1. INRIA Bordeaux Sud-Ouest.
2. LaBRI, Bordeaux INP, CNRS, UMR 5800.
3. Institut des Maladies Neurodégénératives,
Université de Bordeaux, CNRS, UMR 5293.
Bordeaux, France.
orcid.org/0000-0003-0249-2080

Xavier Hinaut
1. INRIA Bordeaux Sud-Ouest.
2. LaBRI, Bordeaux INP, CNRS, UMR 5800.
3. Institut des Maladies Neurodégénératives,
Université de Bordeaux, CNRS, UMR 5293.
Bordeaux, France.
orcid.org/0000-0002-1924-1184

***Abstract*—The modeling of children language acquisition with robots is a long quest paved with pitfalls. Recently a sentence parsing model learning in cross-situational conditions has been proposed: it learns from the robot visual representations. The model, based on random recurrent neural networks (i.e. reservoirs), can achieve significant performance after few hundreds of training examples, more quickly that what a theoretical model could do. In this study, we investigate the developmental plausibility of such model: (i) if it can learn to generalize from single-object sentence to double-object sentence; (ii) if it can use more plausible representations: (ii.a) inputs as sequence of phonemes (instead of words) and (ii.b) outputs fully independent from sentence structure (in order to enable purely unsupervised cross-situational learning). Interestingly, tasks (i) and (ii.a) are solved in a straightforward fashion, whereas task (ii.b) suggest that that learning with tensor representations is a more difficult task**

***Index Terms*—Reservoir Computing, Echo State Networks, Unsupervised Learning, Cross-Situational learning, Robot, Language Learning, Concept formation and symbol grounding/emergence, Language acquisition, Language and semantic reasoning.**

I. Introduction

Numerous studies investigate various aspects of children language acquisition. During the first year of life, children already combine numerous steps [1], as well as various mechanisms such as sensorimotor imitation learning [2] within their language acquisition process. Developmental psychology studies focus on the developmental aspects of how different steps enable to stack upon one another, on models investigating how the various mechanisms can be combined to manage all these steps one after another [3], and on subsequent integration of such models in robots [4]. Some psychological studies investigate the bases of interaction created between an infant and its caregiver [5], necessary to root verbal communication and the formation of abstract symbols [6]. Harnad stated the famous *Symbol Grounding Problem* [7] and shown the importance for a system manipulating symbol to "anchor" their meanings to "raw" perceptions: the so-called symbol grounding that several studies experimented in robots [8], [9]. From a developmental perspective, a bottom-up approach that let symbols emerge seems relatively feasible and more appropriate [10]. Within the family of symbol grounding research, some studies investigate how biologically plausible neural-based mechanism could model the language processing in brain [11]. However, few studies directly investigate how to model developmental aspects of symbol grounding learning. Thus, our study is motivated towards this research axe, extending a recent model proposed by Juven et al. [12] for language grounding with reservoir computing.

II. Related work

Our study is inspired by the work of Juven [12] on language acquisition model with reservoir computing via cross-situational learning. In his work, Juven adapted the ResPars model extensively described in [11] jointly with its neurobiological foundations. The model is able to capture semantics in a complex sentence by co-occurrences of words and perform sentence-level comprehension. Juven's is trained with teacher signals provided by a simulated vision. The model is trained via *cross-situational learning*: sentences given at input only describe partially the simulated vision (i.e. object features or entire objects can be omitted in the sentence), while teacher signals contain information on complete visual perception. Therefore, Juven's model could theoretically enable robots to learn language basis (i.e. description of objects and their features given non trivial sentences) by themselves without any supervision.

Nevertheless, Juven's model has limits. The first limit is that words are used as input to the model. Some works have shown that words could be extracted directly from speech (i.e. *word discovery* [13]). However due to the natural ambiguity of speech, contextual information is often necessary to correctly segment words from continuous speech. In addition, developmental studies have shown that children first discriminate phonemes before learning to recognize words and their meanings [1]. Thus, whether used for Human-Robot Interaction applications or children-linguistic developmental research, word acquisition from raw audio signals would be a first prerequisite for Juven's model. Importantly, Juven's output representation depends on the number and the chronological order in which objects are described. Thus, teacher signals still depend on

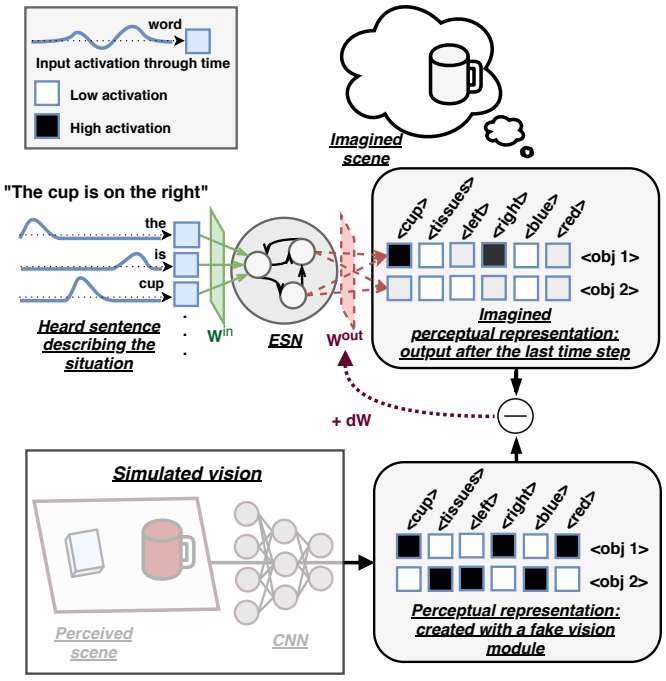

Fig. 1. The figure represents an entire cross-situational learning pipeline, containing a vision model and a language model. As can be seen below the legend (on the top left), the ResPars model is composed of an Echo State Network (ESN) which receives a sequence of words (i.e. a sentence) as input and is trained to recognize the objects described in the sentence. While training the ESN, only the output weights W_{out} are modified. Visual perception of the scene (described by the sentence) is provided by the vision part and converted into teacher output for the language model. In Juven's work and also our study, simulated vision replaces the real vision model. The output representation for language model proposed by Juven is structured by the chronological order in which objects are described in the sentence. If only one object is described, this object should be represented in $< obj1 >$; in such case simulated vision would still fill in the teacher output with a random object for $< obj2 >$; Juven's model successfully learns to ignore such $< obj2 >$ in such case. Image comes from [12].

sentence structure, even if the model can be trained with cross-situational learning, which prevents the model from learning in a fully unsupervised fashion.

Our study focuses on exploring the same architecture for language acquisition model proposed by Juven [12] (figure 1), under the developmental point of view, as well as different conditions to overcome the limits of Juven's work. We investigate the model for: (a) its ability to generalize from short single-object sentences to longer double-object sentences, (b) its capability to handle sequences of phonemes, (c) a new output representation independent of the order that objects are described in the sentence, thus allowing fully unsupervised cross-situational learning.

III. Methods

A. Reservoir computing and FORCE learning

Reservoir Computing (RC) is a paradigm for training Recurrent Neural Networks (RNN). It is simple, efficient and only requires low-computational resources. Originally, it was aimed both to model brain cortical areas in the computational neuroscience field, and to overcome gradient vanishing problem suffered by RNNs trained with Back Propagation Through Time (BPTT) in the machine learning field. In the RC paradigm, RNN hidden units and input weights are immutable and randomly initialized; only output weights (i.e. readout layer) are trainable. Training a RNN within RC paradigm only requires to update readout layer instead of training the whole network with BPTT. Figure 1 shows how RNN model with RC paradigm is used in the work of Juven.

In our study, we focus on Echo State Network [14] (ESN), an instance of RC paradigm. Equations (1) and (2) compute internal states and outputs of the ESN respectively for each time step. In general, each input "step" of the sequence is projected to a high-dimensional space inside the reservoir via the input weights and then the recurrent internal connections: this captures non-linear relations between different parts of the input sequence. Then, the readout layer maps the internal space to output space, where only interesting relations are kept during training. In the equations below: $\mathbf{W}$, $\mathbf{W}^{\text{in}}$ and $\mathbf{W}^{\text{out}}$ are the reservoir internal, input and output weights matrices; $\mathbf{x}_t$, $\mathbf{u}_t$ and $\mathbf{y}_t$ denote internal states, input and output at time t.

$$\mathbf{x}_t = (1-\alpha)\mathbf{x}_{t-1} + \alpha \tanh\left(\mathbf{W}\mathbf{x}_{t-1} + \mathbf{W}^{\text{in}}\mathbf{u}_t\right) \quad (1)$$

$$\mathbf{r}_t = \begin{pmatrix} 1 \\ \mathbf{x}_t \end{pmatrix} \qquad \mathbf{y}_t = \mathbf{W}^{\text{out}}\mathbf{r}_t \quad (2)$$

Similarly as how human learns language, we use an iterative and gradual online learning. In our study, we use FORCE learning [15], an online learning method for reservoirs. The general idea of FORCE is to fit the readout layer weights so as to keep model prediction error small and stable with any new data. FORCE learning employs a modified version of Recurrent Least Square (RLS) algorithm to update readout layer $\mathbf{W}_t^{\text{out}}$ for each time step. Equations (3) and (4) computes e_t^- (i.e. prediction error at time t) and $\mathbf{P}_t$ (i.e. correlation matrix between current state and history states) respectively. Readout weights are updated via equation (5) at the end of each time step.

$$e_t^- = \mathbf{y}_t^{\text{predict}} - \mathbf{y}_t^{\text{target}} = \mathbf{W}_{t-1}^{\text{out}}\mathbf{r}_t - \mathbf{y}_t^{\text{target}} \quad (3)$$

$$\mathbf{P}_0 = \frac{\mathbf{I}}{\alpha} \qquad \delta\mathbf{P}_t = -\frac{\mathbf{P}_{t-1}\mathbf{r}_t\mathbf{r}_t^\top\mathbf{P}_{t-1}}{1+\mathbf{r}_t^\top\mathbf{P}_{t-1}\mathbf{r}_t} \quad (4)$$

$$\mathbf{W}_0^{\text{out}} = \mathbf{0} \qquad \delta\mathbf{W}_t^{\text{out}} = -e_t^-\mathbf{r}_t^\top\mathbf{P}_t \quad (5)$$

Since FORCE learning uses RLS under the hood, regularization parameter α plays an important role. Used for initializing matrix $\mathbf{P}$ and acting as learning rate, α has to be selected appropriately for specific task. Small α results in fast learning, but also makes weights change so quickly that the model may become unstable. By contrast, a model with large α has a slower learning rate, but sometime unable to keep up with rapid change in data dynamics.

To implement reservoir models in our study, we used *ReservoirPy* v0.2 [16]. It is an efficient library to design ESNs, which already supports offline and online training, as well as computational parallelization, fast reservoir initialization and other necessary utilities, such as optimized parameters search with hyperopt [17]. For supporting FORCE learning, ReservoirPy is extensively used in our study to facilitate implementation effort.

B. Corpus and teacher output generation

Corpus contains sentences given to the model at input. In our work, the corpus are generated under our designed grammar (figure 2). Comparatively to the grammar introduced in Juven's work, we use 4 objects, 6 colors and 3 positions (instead of 4, 4, 4 respectively). In addition, we add one more form for sentences: *there is the* $(< color >?) < object > on$ *the* $< position >$.

Teacher outputs are expected outputs after processing input sentences, used to train the model. In our study, teacher output represents exactly objects described in the sentence. More concretely, if the sentence describes an object without color, teacher output shows an object with *unknown-color* feature. We had also performed experiments without this *unknown* category on our proposed model, but we did not manage to obtain good enough performances. Further solutions would be explored in future work.

OBJ → **cup** | **bowl** | **apple** | **spoon**
COL → **red** | **orange** | **yellow** | **green** | **blue** | **magenta**
POS → **left** | **middle** | **right**
THE → **a** | **the**
THIS → (**this** | **that**)
SENTENCE-1-OBJ → THIS **is** THE (COL)? OBJ
| THE OBJ (**on the** POS)? **is** COL
| THE (COL)? OBJ **is on the** POS
| **there is** THE (COL)? OBJ **on the** POS
| **on the** POS (**there**)? **is** THE (COL)? OBJ
SENTENCE-2-OBJ → SENTENCE-1-OBJ
| SENTENCE-1-OBJ **and** SENTENCE-1-OBJ

Fig. 2. Grammar used to generate the corpus. The grammar can generates sentences describing one or two objects depending on the scenario. The total number of different sentences that could be generated is 952976 ($= 976^2$).

C. Phoneme inputs

One of the limits in Juven's model [12] is the use of words as input, because it is not natural for HRI applications, as well as for modeling linguistic development in children. Meanwhile, phoneme seems more biologically plausible, since children develops their capacity to recognize phonemes[1] before learning words.

[1]A phone is the smallest acoustic unit in human language, and a phoneme is the smallest invariant unit (a phoneme can include several variants of phones: e.g. different variants of pronouncing word "a").

In order to explore if reservoir model for language grounding is capable to deal with phonemes, we experimented on Juven's model. Words in the corpus are converted into sequences of phonemes based on the Carnegie Mellon University word-phoneme dictionary (CMUdict v0.07)[2] as described in [18]. Then, using the same way for input encoding as with words (i.e. one hot encoding for each phoneme), we train the model on corpus of phonemes and evaluate its performance.

D. Generalization property

Experimenting on Juven's model, we discovered an interesting property showing the developmental plausibility of reservoir-based language models. In general, the model, trained on single-object sentences (e.g. "*there is a red cup on the left*"), is capable of recognizing object features described in longer sentences. In our case, we tested with double-object sentences, where each sentence is a concatenation of 2 single-object sentences and the word "*and*" (e.g. "*the red apple is on the right and there is blue bowl on the left*"). Given that all object features are already encountered by the model when it is trained on single-object sentences, the model can still recognize those features and ignore the words like "*and*", which do not play an important role in describing objects, in longer sentences even though the model has never seen those sentences before.

Despite having this very interesting property, the output representation in Juven's model depends on the number of objects predefined in the scene, which is tightly coupled with the sentences describing it. If the model was trained on single-object sentences, it can only return one object at output. Thus, the object features recognized in double-object sentences are mixed up and hardly separable in order to extract them from the output representation.

E. Output representation

Important limits in Juven's model stem from its output representation, which is dependent on each utterance and does not allow to train the model in a purely unsupervised fashion. Hence, we propose a new output representation which is independent of the order in that objects are mentioned, as well as the number of objects in the scene.

In our experiments, each object has 3 features: category (e.g. cup), color (e.g. red) and position (e.g. right). Thus, we use a 3D tensor to represent output, where each axis represents one feature. Provided that there is no two identical objects (with the exact same features) in a sentence, this output can represent all objects mentioned in the sentence at once. Further work will explore if updated output representations can handle cardinality (i.e. number of the same objects) in the output activation level.

In our study, we choose to use a sparse encoding scheme for output, so as to speed-up computations. In our proposed representation, each *cell* (i.e. smallest unit inside the tensor) identifies an object with all of its features (i.e. the coordinates

[2]CMUdict can be found at http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

of the cell). The teacher output is, therefore, generated by firstly initializing the whole tensor with zeros, then setting all the cells corresponding to objects described in the sentence with 1. This is the so-called *object activated* (or *cell activated*) encoding scheme. Figure 3 shows an example of this scheme.

Consistent to this encoding scheme, given the number of objects in the scene, the results are extracted from the tensor representation by selecting that same number of cells with highest values among others.

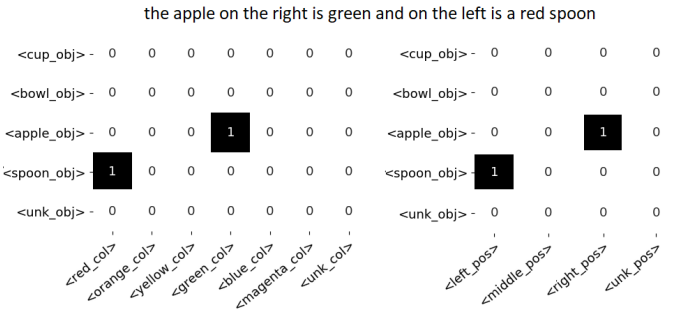

Fig. 3. Example of teacher output for provided sentence with our proposed output representation. The 3D tensor is projected alongside 2 axis, showing the association of object features described in the sentence.

In our study, we also explored other encoding schemes for tensor representation, though the model does not achieve significant performance with them.

- *Feature activated* (or *plan activated*): the tensor is initialized with zeros. Then, for each feature mentioned in the sentence, a "weight" is added to the values of the cells representing objects sharing that feature in common (i.e. the cells on a same "feature plan"). Finally, the same weight is added for the cells representing exactly the objects described in the sentence.
- *Lateral inhibition*: taking inspiration from Self-Organising Maps [19], each cell representing exactly the object is positively activated, while all the plans sharing these cells are negatively activated.

F. Evaluation

The *valid evaluation metric* as presented in the work of Juven is adopted to measure how well our model performs. An output is considered valid when it contains at least all objects and their features described in the sentence. For example, a valid output for sentence "*there is a cup on the left*" can be $(< cup >, < left >, < red >)$.

G. Experiments

This part summarizes the experiments conducted during our study. For all experiments, the model is trained with FORCE learning and readout layer is updated at the end of each sentence. Optimial values for reservoir hyparameters were obtained with hyperopt beforehand. In the experiments where the model's performance is measured (experiment II, III), test set contains sentences that the model has never encountered during training.

This section is structured as following: (i) experiment on the model capability to generalize from single-object to double-object sentences, (ii) experiment on model's performance with phonemes input, (iii) experiments on model's performance with tensor output representation under various conditions, including: different number of sentences for training, different reservoir sizes. The experiment derivation tree is listed below, based on different studies on reservoir language model

- Input representation
  - word (experiment I, II, III)
  - phoneme (experiment II)
- Output representation
  - Juven's model (experiment I, II)
  - tensor
    - object activated (experiment III)
    - feature activated
    - lateral inhibition

1) Experiment I: Generalization property: In this experiment, we evaluated Juven's model for its capability to generalize from single-object to double-object sentences. The model is trained on 1000 single-object sentences and then tested on double-object sentences. The output is plotted to show how the object features are activated at output during the whole sentence. In the plot, y axis illustrates the activation intensity, while x axis shows the words in the sentence. The plot is provided and discussed in subsection IV-A.

2) Experiment II: Phonemes input: In this experiment, we compare the performance of Juven's model on input with phonemes and words. The model is evaluated on 2 scenarios: single-object and double-object corpus. According to the grammar described in figure 2, there are much less single-object sentences than double-object sentences. Thus for single-object corpus, we used a same size of 400 sentences for train and test set, while in the second scenario corpus size increases to 1000. Results are provided in subsection IV-B.

3) Experiment III: Tensor representation: In this experiment, we evaluate the model's performance with tensor representation. With hyperopt, the optimal values for reservoir hyperparameters are: spectral radius λ = 0.3, regularization coefficient α = 10^{-8}, leak rate = 0.15, input scaling = 1.0, reservoir sparsity = 0.1. The model is then trained and evaluated on double-object corpus, using those optimal values for reservoir hyperparameters. In this experiment, we investigate the model's performance with respect to (w.r.t)

(a) *Number of sentences for training*: The reservoir size is fixed at 1000 neurons. The model is trained on 10,000 sentences in total. For each step of 1000 sentences, the model is tested on 300 sentences. The model's performance is measured and plotted. Results are provided and discussed in subsection IV-C. The optimal number of sentences for training is used in the next experiment.

(b) *Reservoir size* (i.e. number of recurrent hidden units, representing memory capacity and non-linear computational

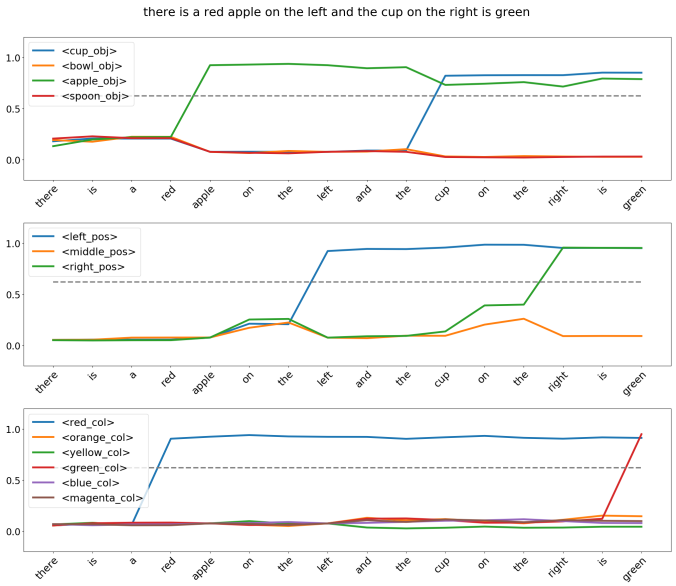

Fig. 4. Capability to generalize from single-object to double-object sentences. Y axis: activation intensity, X axis: words in the sentence.

	Phoneme input	Word input
1-object scenario (train = 400 / valid = 400)	0.0 (±0.0)	0.0 (±0.0)
2-object scenario (train = 1000 / valid = 1000)	0.6 (±0.39)	1.6 (±0.66)

TABLE I
ERROR ON VALID METRIC FOR SINGLE-OBJECT AND DOUBLE-OBJECT SCENARIOS WITH PHONEMES OR WORDS INPUT

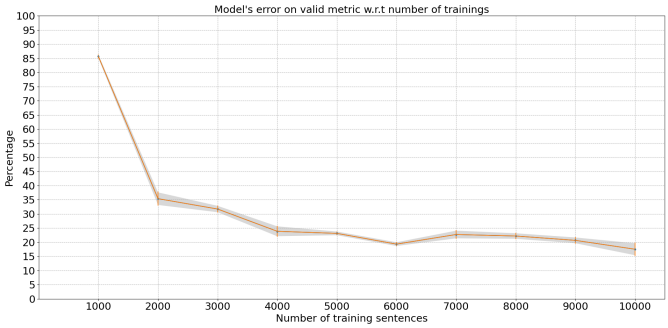

Fig. 5. Error (evaluated with valid metric) of model w.r.t number of sentences for training.

power of reservoir models): The model is trained on 5,000 sentences and also tested on other 300 sentences. The performance is plotted and discussed in subsection IV-D.

IV. Results

A. Experiment I: Generalization property

Figure 4 shows how the model behaves with double-object sentences when being trained on single-object sentences only. The plot shows the activation of object features at output. From the figure, we can observe that correct object features are activated when the model encounters the words describing them. Since the teacher output is only given to the model at the end of the sentence, this interesting property indicates that the model is able to intrinsically "compare" the sentences given at input and teacher signals at output to learn to know which word describes which object feature. Moreover, the model never encountered double-object sentences during training, but it can still trigger correct features of second object at output. Thus, this behavior suggests that the model is able to learn and "remember" the mapping of words and object features that those words describe from simple single-object sentences and generalize to longer sentences. Consequently, when a double-object sentence is given to the model, correct features are activated corresponding to the words describing them.

B. Experiment II: Phonemes input

Table I compares model error evaluated with valid metric (subsection III-F) on phonemes and words input. As we can observe from the table, the model has similar performance with phonemes as well as with words: both achieves an error of around 1%.

C. Experiment IIIa: Number of training sentences

Figure 5 shows the evolution of model's error rate w.r.t the number of sentences used for training. The plot shows a steep decrease in error rate from 85% to 35% when the number of sentences increases from 1000 to 2000. The trend continues, but more slightly, to 5000 sentences before fluctuating around 17% of error rate. Since our model needs more sentences for training in comparison with Juven's model before reaching a reasonable performance, it suggests that learning with tensor representation is a more difficult task. Based on those results, we select 5,000 sentences to train our models for the rest of our experiments.

D. Experiment IIIb: Number of reservoir hidden units

Figure 6 shows model's error rate in relation to the reservoir size. As we can see from the plot, a reservoir with more neurons performs better (i.e. having a lower error rate). Indeed, the number of neurons is considered as memory capacity and non-linear computational power of the reservoir. As the experiments IIIa suggest that learning with tensor output is more difficult, more neurons can help to improve the global performance of reservoir models. However, the performance does not improve much when the number of neurons exceeds 2000, this may suggest that other hyperparameters have to be modified so that the model can achieve better performance. Hence, a reservoir of 2000 neurons and trained with 5,000 sentences probably achieves its best performance in our study.

Supplementary experiments and results can be found at https://github.com/neuronalX/DinhHinaut2020-ICDL

V. Discussion

Despite developmental language evidences and robot learning progress [4], the modeling of children-like language ac-

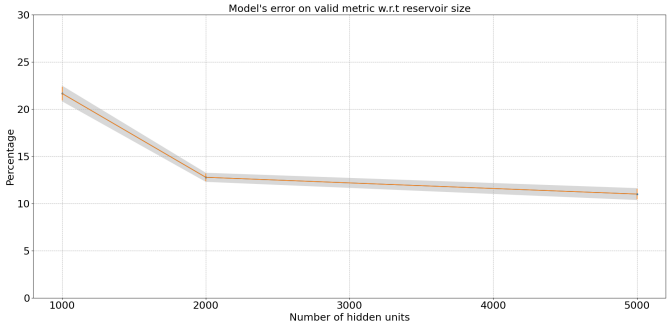

Fig. 6. Error (evaluated with valid metric) of model w.r.t reservoir size.

quisition in robots seems still in an early stage. Juven et al. [12] proposed a sentence parsing model learning in cross-situational conditions (a new version of ResPars model [11]): it learns from the visual scene a robot perceives. In this study, we investigated the developmental plausibility of this model and proposed new representations for outputs based on tensors. These new representations did not rely on the input sentence structure, enabling the model to learn purely from unsupervised cross-situational learning.

The reservoir-based model is able to learn on the new proposed representations, even though the task becomes more difficult. Indeed, twice more neurons and a bigger training corpus are needed to obtain significant performance. We verified this increased task difficulty with extensive hyperparameter search using *hyperopt* [17] and the graphical tool provided in *ReservoirPy* [16]. The fact that the representation proposed in Juven's model is dependent on the sentence structure (i.e. the order in which words appear) is not specific to the task, nor to the ResPars model: it is a common way of representing the role of words in a sentence like in Semantic Role Labelling. Thus, searching solutions for the sentence-structure independence is not specifically limited to our study, but would be useful in general for developmental language models.

Interestingly, we discovered two abilities of Juven's model, which were confirmed also for the tensor representations. Firstly, the model is able to process sentences composed of sequence of phonemes with the same performance as with words, thus making the models not dependent on word segmentation and categorization. Secondly, the model is able to generalize to double-object, or probably multiple-object sentences when it is only trained with single-object sentences. These new abilities demonstrate more developmentally plausible mechanisms and behaviors. In particular, this generalization capability indicates that the ResPars model has intrinsic properties for scaffolding learning.

The new tensor representations is our first attempt to reach a version of ResPars that could learn in fully unsupervised cross-situational learning conditions, in order to approach how humans, especially children, learn language. Several tracks can be followed to pursue this work. For example, experiment with LSTMs on similar tasks to see if the tensor representations are also more difficult to learn for LSTMs Finally, we want to use a hierarchical-task reservoir [20] in order to decompose this difficult task in two sub-tasks. A first reservoir would associate sequence of phonemes to individual concepts/features (i.e. object features), and a second reservoir would merge the concepts together within a complete object representation.

Currently, as a follow-up to this study, we are investigating a new representation, which is also independent of the sentence structure but has a much smaller size than tensor representation. More importantly, this novel representation scales because its size is proportional to the number of object features, and because of its ability to exploit effectively and extensively the generalization property.

REFERENCES

[1] P. K. Kuhl. Early language acquisition: cracking the speech code. *Nature Reviews Neuroscience*, 5(11):831–843, Nov 2004.

[2] S. Pagliarini et al. Vocal Imitation in Sensorimotor Learning Models: a Comparative Review. HAL Preprint hal-02317144, October 2019.

[3] E. Dupoux. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59, April 2018.

[4] A. Cangelosi et al. Integration of action and language knowledge: A roadmap for developmental robotics. *IEEE Transactions on Autonomous Mental Development*, 2(3):167–195, 2010.

[5] I. Nomikou et al. Taking up an active role: emerging participation in early mother–infant interaction during peekaboo routines. *Frontiers in psychology*, 8:1656, 2017.

[6] J. Raczaszek-Leonardi et al. Language development from an ecological perspective: Ecologically valid ways to abstract symbols. *Ecological Psychology*, 30(1):39–73, January 2018.

[7] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990.

[8] M. Spranger et al. Open-ended procedural semantics. In *Language grounding in robots*, pp. 153–172. Springer, 2012.

[9] X. Hinaut and M. Spranger. Learning to parse grounded language using reservoir computing. In *Proc. of ICDL-Epirob*. IEEE, August 2019.

[10] T. Taniguchi et al. Symbol emergence in robotics: a survey. *Advanced Robotics*, 30(11-12):706–728, 2016.

[11] X. Hinaut and P. Dominey. Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing. *PLoS ONE*, 8(2):e52946, 2013.

[12] A. Juven and X. Hinaut. Cross-Situational Learning with Reservoir Computing for Language Acquisition Modelling. In *Proc. of IJCNN 2020)*.

[13] T. Taniguchi et al. Double articulation analyzer with deep sparse autoencoder for unsupervised word discovery from speech signals. *Advanced Robotics*, 30(11-12):770–783, April 2016.

[14] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report 148, German National Research Center for Information Technology GMD, Bonn, Germany, 1 2001.

[15] D. Sussillo and L. F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, Aug 2009.

[16] N. Trouvain et al. ReservoirPy: an Efficient and User-Friendly Library to Design Echo State Networks. In *ICANN*, 2020. https://github.com/neuronalX/reservoirpy.

[17] J. Bergstra et al. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in Science Conference*, pp. 13–20, 2013.

[18] X. Hinaut. Which input abstraction is better for a robot syntax acquisition model? phonemes, words or grammatical constructions? In *Proc. of ICDL-Epirob*. IEEE, 2018.

[19] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[20] L. Pedrelli and X. Hinaut. Hierarchical-task reservoir for anytime pos tagging from continuous speech. In *Proc. of IJCNN 2020*, 2020.

[21] T.-Y. Lin et al. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

SUPPLEMENTARY MATERIAL

We made similar experiments as presented above with our model (i.e. reservoir model with tensor representation) with the same conditions, as well as under various changes in teacher output generation and input types. Same values for hyperparameters (obtained from hyperopt and experiments in subsection III-G3 on tensor representation) are used to favor comparison. The experiments provide interesting results that may be extended in future work.

A. Experiment I-sup: Generalization from 1 to 2 objects

In this experiment, we investigate the capability of our model to generalize from single-object to double-object sentences. The output of our model is a 3D object-feature tensor with object-activated encoding scheme. In this experiment, we selectively project the output tensor alongside individual feature axes to observe how the model behaves with the features described in the sentence. Results are shown in figure 7. In addition, we also project the tensor on a combination of 2 features, in order to see whether the features are correctly associated together as described in the sentence. The projection is done with max operation (i.e. taking the max values over all cells that represent the objects sharing the shame features to be projected on). The results are given in figure 8.

Figure 7 shows the same output activation plot as in experiment 4, but with tensor representation. The model is also trained on single-object sentences before being tested with double-object sentences. As we can observe from the plot, our model has the same generalization property, but the activation intensity is much lower. Besides, unrelated object features are slightly positively triggered at non-functional words (i.e. words that do not represent any object features), before being negatively triggered at functional words. Those differences are subjected to further study for better understanding on how our proposed model behaves.

Figure 8 shows a more detail view on the output representation of our model with the same sentence in figure 7. The figure shows that ($< cup\_obj >$, $< right\_pos >$, $< green\_col >$) is the most activated cell, correctly corresponding to *the green cup on the right*. However, the model did not recognize *the red apple on the left*. Indeed, cells associated $< apple\_obj >$ and $< left\_pos >$ features only have a maximum value of 0.1, whereas most activated cells associated to $< apple\_obj >$ come with $< right\_pos >$ feature, but they only have a slightly higher intensity at 0.11. This implies that future research to improve the model's performance on tensor representation should focus on resolving correcting these kinds of wrong association.

B. Experiment III-sup: Number of object categories

We investigate the scalability of model with tensor representation by controlling the number of objects (i.e. object names). Indeed, more number of objects result in larger corpus vice versa. In this experiment, we evaluated the error of our proposed model w.r.t different number of object categories, varying from 5 to 30. In addition, we also use a theoretical

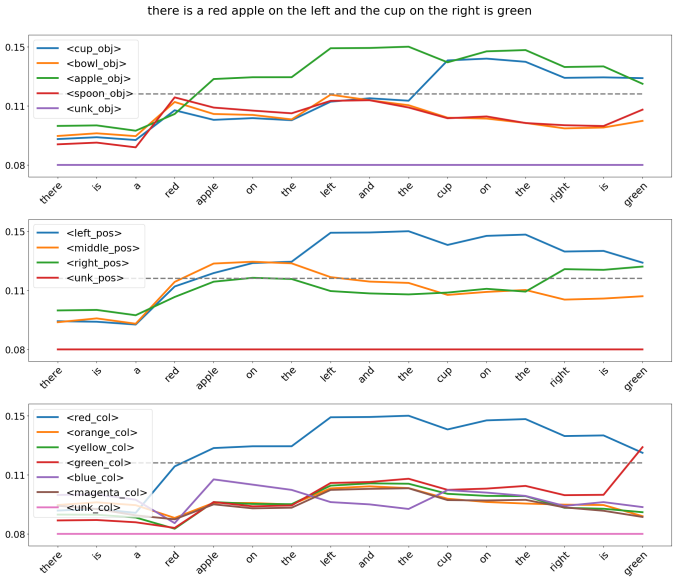

Fig. 7. Generalization capacity from 1-object to 2-object sentences of our model. Y axis: activation intensity, X axis: words in the sentence.

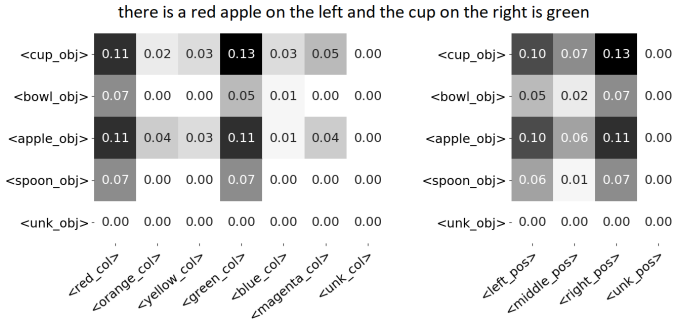

Fig. 8. Output tensor projected onto 2 axis, showing the characteristics associated with objects described in the same sentence as in Figure 7.

model proposed in the work of Juven [12] as baseline for comparison. The error of such model is computed via equation (6), provided that the total number of potential colors and positions are predefined (i.e. $n_{\text{col}} = 6$ for red, orange, yellow, green, blue, magenta and $n_{\text{pos}} = 3$ for left, middle, right). We adapt the equation with the grammar used for generating our corpus.

$$err_{\text{th}}(n_{\text{obj}}, n_{\text{train}}) = \left(1 - \frac{1}{244 \times n_{\text{obj}}}\right)^{2 \times n_{\text{train}}} \quad (6)$$

Figure 9 illustrates the performance of our model with different number of object categories (i.e. orange solid line) in comparison with the error of theoretic model (i.e. grey dashed line). The plot shows that our model is still not as good as the theoretical model. However, the error evolution line has similar trend w.r.t number of object categories.

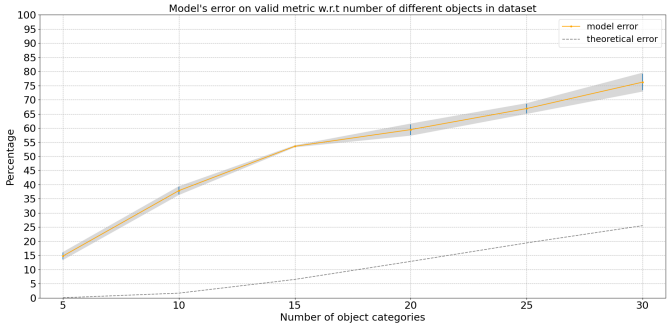

Fig. 9. Valid error metric of model w.r.t number of object categories.

C. Phonemes input

We conducted the same experiment on phonemes with our proposed model. The result demonstrates that our model has the same level of performance with phonemes as with words, thus can be used to study linguistic development in children, as well as applied for other HRI applications.

Result with words: 23.1 ± 0.72

Result with phonemes: 19.0 ± 1.52

D. Tensor representation without unknown features

We studied whether it is possible to adapt the tensor representation without *unknown* values on the feature axes when generating teacher outputs, so that the model allows fully unsupervised cross-situational learning. Our naive approach is to set cells representing objects combined with all possible values of the unknown feature to 1. Literally, it is equivalent to telling the model that all colors are possible for a cup, when the sentence describes that cup with unknown color.

However, the result shows that model has higher error rate. Indeed, teacher outputs generated in that way are unbalanced, because the number of cells set to 1 depends on the utterance and is influenced by the statistics of the training data.

Result: 52.0 ± 1.89

E. Feature-activated (plan-activated) encoding scheme

In this experiment, we studied feature-activated encoding scheme for generating teacher output. As described in subsection III-E, this encoding schemes helps generate a teacher output where a cell with higher value has more chance to describe the right object. Indeed, this scheme results in more cells with non-zero values in the tensor, compared to object-activated encoding scheme, and their valuestr are proportional to the chance that they describe the right objects in the scene.

We can see from the result that the error decreases by a large margin compared to previous experiment on representation without unknown features. Even though the error is still higher than the representation with unknown features and object-activated encoding scheme. This way of encoding teacher output is still interesting for future work, since it allows fully unsupervised learning on our model. However, more study is needed to investigate the model's behaviors under various conditions, for example different weight added for object features and entire objects.

Result: 32.9 ± 1.81

COMPLEMENTARY STUDIES

In this section, we introduce a way to filter the corpus so that the objects described in the sentences are more realistic, as well as experimented the new corpus with Juven's model. Beside, we also tested a possibility to extend Juven's model only by changing teacher output encoding scheme.

F. Vision-biased corpus

We conducted an experiment on Juven's model with a vision-biased corpus to explore further the model's learning capability via cross-situational learning. Exploiting training labels of COCO dataset for object segmentation task [21], we extract and filter only real possible values for object features available in COCO images when generating corpus. Thus, the generated corpus is less equally distributed among object features (e.g. there is no sentence describing a magenta apple), but closer to human utterances. The experiment is conducted to measure the error of Juven's model under the same conditions and reservoir hyperparameters as used in experiment comparing phonemes and words input (subsection III-G2) for the double-object corpus and words input. Comparing the result with table I, the model error is higher when training with this vision-biased corpus. However, an error rate of 2% demonstrates that the model can still perform well under realistic conditions of fully cross-situational learning.

Result: 2.1 ± 0.63

G. Abstract-ranking encoding

As an idea to overcome the limit on sentence-structure dependency of the output representation in Juven's model, we exploit the tensor representation to sort objects in the scene with an arbitrary rank (when the tensor is flatten) before encoded it in teacher output. Thus, the order that objects are encoded in teacher output does not depend on the sentence describing the scene anymore, but on how the tensor representation is designed. Juven's model is trained and evaluated under the same conditions and reservoir hyperparameters as in the experiment above.

The result shows a high error rate. However, the experiment is still interesting, since it demonstrates that learning such an arbitrary ranking is a difficult task for reservoir models. Hence, the lower performance that we obtained with our tensor representation (compared to the original model proposed in Juven's work) is probably due to this new constraint (for the ResPars model) to have an output representation completely independent of the structure of the sentence. This is reassuring in a way, highlighting a potentially inherent issue rather than the consequences of our specific representation choices.

Result: 67.4 ± 0.63